CLAIMS

We claim:

1. A method of parsing a data file into its static and dynamic data components comprising:

setting a threshold value for separating static data components and dynamic data components;

obtaining expiration parameters for each data component of the data file;

comparing each expiration parameter to the threshold value;

identifying as dynamic data components the data components of the data file that have an expiration value lower than the threshold value; and

identifying as static data components the data components of the data file that have an expiration value higher than the threshold value.

2. The method of claim 1 wherein the threshold value is set based on a measurement of a load value of a file server storing the data file.

3. The method of claim 1 wherein the threshold value of a first file server is different than the threshold value of other file servers.

4. A method of sending requested data to a client from an intelligent file server storing a file comprising:

receiving a file request from the client, the file including static data components and dynamic data components;

determining whether the client is intelligent;

dividing static data components into static data packets and dividing dynamic data components into dynamic data packets; and

transmitting the static data packets and the dynamic data packets to an intelligent client.

5. The method of claim 4 wherein the intelligent file server stores in at least one memory location a copy of the requested file and a copy of the requested file divided into its static and dynamic components.

6. The method of claim 4 wherein the intelligent file server stores in at least one memory location a copy of the requested file and creates a copy of the requested file divided into its static and dynamic components and create upon receiving the file request.

7. The method of claim 4 further comprising:

identifying routes between the file server and the intelligent client that include at least one intelligent hardware; and

transmitting the static data components and the dynamic data components using the route that includes at least one intelligent hardware.

8. The method of claim 4 further comprising:

counting the number of requests from intelligent clients for a particular data file;

broadcasting the static data components of said data file to each intelligent client once a number of requests for a particular file surpasses a set value.

9. The method of claim 8 wherein the static data components broadcasting further comprising:

(a) receiving a data file including static data components and dynamic data components;

(b) specifying a time interval;

(c) parsing the static data components into a plurality of static data blocks;

(d) determining a required number of time slots to send the data file, wherein each of said time slot has a duration substantially equal to said time interval;

(e) allocating to each time slot at least;

(1) a first of said plurality of data blocks; and

(2) optionally one or more additional data blocks, such that said plurality of data blocks is available to a client accessing said data file during any time slot; and

(f) sending said plurality of data blocks based on said allocating step (e).

10. A method for processing data received from an intelligent server comprising:

sending a file request from an intelligent client to the intelligent server, the intelligent server storing a copy of a file divided into its static components and its dynamic components;

receiving static data packets formed by parsing the static data components and dynamic data packets formed by parsing the dynamic data components;

reconstructing the requested file from the received static data packets and dynamic data packets.

11. The method of claim 10 including the step of receiving static data from a broadcast, further comprising:

(a) receiving a selection of a data file during a first time slot;

(b) receiving at least one data block of said data file during a second time slot; and

(c)during a next time slot:

(1) receiving any data block not already received;

(2) recombining the received data blocks to form the data file; and

(3) repeating step (c) until all data blocks of said data file have been received and recombined.

12. The method of claim 11 wherein further comprising:

performing an error check; and

sending a request to resend the data blocks generating the error check.

13. The method of claim 11 wherein the step of recombining of the data blocks does not start until after all the data files have been received.

14. A method of creating an intelligent protocol data packet comprising:

including identifying information identifying the intelligent data protocol.

15. The method of claim 14 wherein said identification information is included in a header portion of the data packet.

16. The method of claim 15 wherein the data packet is created based on TCP/IP protocol.

17. The method of claim 16 wherein the header portion is the TCP information portion.

18. The method of claim 14 wherein said identification information is included in the data portion of the data packet.

19. The method of claim 18 wherein the data packet is created based on TCP/IP protocol.

20. A method of servicing a data request to a file server from a client connected to the network through a network connection providing server comprising:

receiving a request for a file from the client directed to the file server;

forwarding the request to the file server;

receiving the data from the file server;

storing a copy of a static portions of the received data;

transmitting at least a dynamic data portion of the received data to the client.

21. The method of claim 20 further comprising:

checking a memory related to the network connection providing server for the presence of a first group of data corresponding to the static portions of the requested file;

determining whether the file server is an intelligent file server;

sending the first group of data to the client;

sending a request to the file server for a second group of data not saved in the memory;

receiving the second group of data from the file server; and

sending at least the second group of data to the client.

22. A method of responding to a file request from an intelligent client by an intelligent network connection providing server comprising:

receiving a file request from a client;

determining if the client is intelligent;

sending a static data to the intelligent client;

forwarding a request for at least a dynamic data related to the requested file to an intelligent server storing the requested file;

receiving at least the dynamic data ; and

transmitting at least the dynamic data to the client.

23. The method of claim 22 wherein the client is not intelligent, further comprising:

recombining the received static data for the requested file and the received dynamic data for the requested file; and

repacketing the requested file before transmitting the requested file to the client.

24. The method of claim 22 further comprising:

counting a number of requests from intelligent clients for a particular data file;

broadcasting the static data for the requested file once the number of requests for a particular file surpasses a set value.

25. The method of claim 24 wherein broadcasting the static data further comprising:

(a) specifying a time interval;

(b) parsing the static data into a plurality of static data blocks;

(c) determining a required number of time slots to send the static data, wherein each of said time slot has a duration substantially equal to said time interval;

(d) allocating to each time slot at least;

(1) a first of the plurality of static data blocks; and

(2) optionally one or more additional static data blocks; and

(e) sending the plurality of static data blocks based on said allocating step (d).

26. A method of operation for a system for transmitting and receiving data within a network comprising:

dividing a file into dynamic data and static data by comparing at least one expiration parameter corresponding to each data part to a set threshold value;

transmitting a request for a file from an intelligent client;

receiving the file request by an intelligent server;

transmitting at least the dynamic data components of the requested file to the intelligent client; and

recombining the static components and the dynamic components of the requested file.

27. The method of claim 26 further comprising:

identifying routes between the intelligent file server and the intelligent client that include at least one intelligent hardware; and

transmitting the static data components and the dynamic data components using the route that includes at least one intelligent hardware.

28. The method of claim 26 further comprising:

counting the number of requests from intelligent clients for a particular data file;

broadcasting the static data components of said data file to each intelligent client once a number of request for a particular file surpasses a set value.

29. The method of claim 28 wherein the static data broadcasting steps further comprises:

(a) receiving a static data blocks of a data file;

(b) specifying a time interval;

(c) parsing said data file into a plurality of data blocks based on said time interval such that each data block is displayable during said time interval;

(d) determining a required number of time slots to send said data file, wherein each of said time slot has a duration substantially equal to said time interval;

(e) allocating to each time slot at least;

(1) a first of said plurality of data blocks; and

(2) optionally one or more additional data blocks, such that said plurality of data blocks is available in sequential order to a client accessing said data file during any time slot; and

(f) sending said plurality of data blocks based on said allocating step (e).

30. The method of claim 26 wherein the intelligent file server stores a copy of the requested file separated into its static data components and its dynamic data components and a copy of the requested file that is not separated into its dynamic data components and static data components.

31. A method of sending a file over a network from an originating node to a requesting client, comprising:

receiving a request for the file at an intermediate node;

providing a time insensitive component of the file to the client from the intermediate node; and

providing a time sensitive component of the file to the client from the originating node.

32. The method of claim 31 further comprising:

determining whether or not components of a file are time sensitive or time insensitive based on a comparison between a threshold value and a defined time value for each of the components.

33. The method of claim 32 wherein the threshold value is dependent on the load of the originating server.

34. The method of claim 31 further comprising the steps of:

determining at the intermediate node whether or not the client is adapted to receive time insensitive and time sensitive components; and

if the client is not so adapted,

assembling the file from the time sensitive and time insensitive components at the intermediate node; and

transmitting the assembled file to the client from the intermediate node.

35. A computer program product embodied in a computer readable medium for parsing a data file into its static and dynamic data components comprising:

code means for setting a threshold value for separating static data components and dynamic data components;

code means for obtaining expiration parameters for each data component of the data file;

code means for comparing each expiration parameter to the threshold value;

code means for identifying as dynamic data components the data components of the data file that have an expiration value lower than the threshold value; and

code means for identifying as static data components the data components of the data file that have an expiration value higher than the threshold value.

36. The computer program product of claim 35 wherein the threshold value is set based on a measurement of a load value of a file server storing the data file.

37. The computer program product of claim 35 wherein the threshold value of a first file server is different than the threshold value of other file servers.

38. A computer program product embodied in a computer readable medium for sending requested data to a client from an intelligent file server storing a file comprising:

code means for receiving a file request from the client, the file including static data components and dynamic data components;

code means for determining whether the client is intelligent;

code means for dividing static data components into static data packets and dividing dynamic data components into dynamic data packets; and

code means for transmitting the static data packets and the dynamic data packets to an intelligent client.

39. The computer program product of claim 38 wherein the intelligent file server stores in at least one memory location a copy of the requested file and a copy of the requested file divided into its static and dynamic components.

40. The computer program product of claim 38 wherein the intelligent file server stores in at least one memory location a copy of the requested file and creates a copy of the requested file divided into its static and dynamic components and create upon receiving the file request.

41. The computer program product of claim 38 further comprising:

code means for identifying routes between the file server and the intelligent client that include at least one intelligent hardware; and

code means for transmitting the static data components and the dynamic data components using the route that includes at least one intelligent hardware.

42. The computer program product of claim 38 further comprising:

code means for counting the number of requests from intelligent clients for a particular data file;

code means for broadcasting the static data components of said data file to each intelligent client once a number of requests for a particular file surpasses a set value.

43. The computer program product of claim 42 wherein the static data components broadcasting further comprising:

(a) code means for receiving a data file including static data components and dynamic data components;

(b) code means for specifying a time interval;

(c) parsing the static data components into a plurality of static data blocks;

(d) code means for determining a required number of time slots to send the data file, wherein each of said time slot has a duration substantially equal to said time interval;

(e) code means for allocating to each time slot at least;

(1) a first of said plurality of data blocks; and

(2) optionally one or more additional data blocks, such that said plurality of data blocks is available to a client accessing said data file during any time slot; and

(f) code means for sending said plurality of data blocks based on said allocating step (e).

44. A computer program product embodied in a computer readable medium for processing data received from an intelligent server comprising:

code means for sending a file request from an intelligent client to the intelligent server, the intelligent server storing a copy of a file divided into its static components and its dynamic components;

code means for receiving static data packets formed by parsing the static data components and dynamic data packets formed by parsing the dynamic data components;

code means for reconstructing the requested file from the received static data packets and dynamic data packets.

45. The computer program product of claim 44 including the step of receiving static data from a broadcast, further comprising:

(a) code means for receiving a selection of a data file during a first time slot;

(b) code means for receiving at least one data block of said data file during a second time slot; and

(c)during a next time slot:

(1) code means for receiving any data block not already received;

(2) code means for recombining the received data blocks to form the data file; and

(3) code means for repeating step (c) until all data blocks of said data file have been received and recombined.

46. The computer program product of claim 45 wherein further comprising:

code means for performing an error check; and

code means for sending a request to resend the data blocks generating the error check.

47. The computer program product of claim 45 wherein the step of recombining of the data blocks does not start until after all the data files have been received.

48. A computer program product embodied in a computer readable medium for creating an intelligent protocol data packet comprising:

code means for identifying information identifying the intelligent data protocol.

49. The computer program product of claim 48 wherein said identification information is included in a header portion of the data packet.

50. The computer program product of claim 49 wherein the data packet is created based on TCP/IP protocol.

51. The computer program product of claim 50 wherein the header portion is the TCP information portion.

52. The computer program product of claim 48 wherein said identification information is included in the data portion of the data packet.

53. The computer program product of claim 52 wherein the data packet is created based on TCP/IP protocol.

54. A computer program product embodied in a computer readable medium for servicing a data request to a file server from a client connected to the network through a network connection providing server comprising:

code means for receiving a request for a file from the client directed to the file server;

code means for forwarding the request to the file server;

code means for receiving the data from the file server;

code means for storing a copy of a static portions of the received data;

code means for transmitting at least a dynamic data portion of the received data to the client.

55. The computer program product of claim 54 further comprising:

code means for checking a memory related to the network connection providing server for the presence of a first group of data corresponding to the static portions of the requested file;

code means for determining whether the file server is an intelligent file server;

code means for sending the first group of data to the client;

code means for sending a request to the file server for a second group of data not saved in the memory;

code means for receiving the second group of data from the file server; and

code means for sending at least the second group of data to the client.

56. A computer program product embodied in a computer readable medium for responding to a file request from an intelligent client by an intelligent network connection providing server comprising:

code means for receiving a file request from a client;

code means for determining if the client is intelligent;

code means for sending a static data to the intelligent client;

code means for forwarding a request for at least a dynamic data related to the requested file to an intelligent server storing the requested file;

code means for receiving at least the dynamic data ; and

code means for transmitting at least the dynamic data to the client.

57. The computer program product of claim 56 wherein the client is not intelligent and further comprising:

code means for recombining the received static data for the requested file and the received dynamic data for the requested file; and

code means for repacketing the requested file before transmitting the requested file to the client.

58. The computer program product of claim 56 further comprising:

code means for counting a number of requests from intelligent clients for a particular data file;

code means for broadcasting the static data for the requested file once the number of requests for a particular file surpasses a set value.

59. The computer program product of claim 58 wherein broadcasting the static data further comprising:

(a) code means for specifying a time interval;

(b) code means for parsing the static data into a plurality of static data blocks;

(c) code means for determining a required number of time slots to send the static data, wherein each of said time slot has a duration substantially equal to said time interval;

(d) code means for allocating to each time slot at least;

(1) a first of the plurality of static data blocks; and

(2) optionally one or more additional static data blocks; and

(e) code means for sending the plurality of static data blocks based on said allocating step (d).

60. A computer program product embodied in a computer readable medium for operation of a system for transmitting and receiving data within a network comprising:

code means for dividing a file into dynamic data parts and static data parts by comparing at least one expiration parameter corresponding to each data part to a set threshold value;

code means for transmitting a request for a file from an intelligent client;

code means for receiving the file request by an intelligent server;

code means for transmitting at least the dynamic data components of the requested file to the intelligent client; and

code means for recombining the static components and the dynamic components of the requested file.

61. The computer program product of claim 60 further comprising:

code means for identifying routes between the intelligent file server and the intelligent client that include at least one intelligent hardware; and code means for transmitting the static data components and the dynamic data components using the route that includes at least one intelligent hardware.

62. The computer program product of claim 60 further comprising:

code means for counting the number of requests from intelligent clients for a particular data file;

code means for broadcasting the static data components of said data file to each intelligent client once a number of request for a particular file surpasses a set value.

63. The computer program product of claim 62 wherein the static data broadcasting steps further comprises:

(a) code means for receiving a static data blocks of a data file;

(b) code means for specifying a time interval;

(c) code means for parsing said data file into a plurality of data blocks based on said time interval such that each data block is displayable during said time interval;

(d) code means for determining a required number of time slots to send said data file, wherein each of said time slot has a duration substantially equal to said time interval;

(e) code means for allocating to each time slot at least;

(1) a first of said plurality of data blocks; and

(2) optionally one or more additional data blocks, such that said plurality of data blocks is available in sequential order to a client accessing said data file during any time slot; and

(f) code means for sending said plurality of data blocks based on said allocating step (e).

64. The program method program of claim 60 wherein the intelligent file server stores a copy of the requested file separated into its static data components and its

dynamic data components and a copy of the requested file that is not separated into its dynamic data components and static data components.

65. A computer program product embodied in a computer readable medium for sending a file over a network from an originating node to a requesting client, comprising:

code means for embodied in a computer readable medium for receiving a request for the file at an intermediate node;

code means for providing a time insensitive component of the file to the client from the intermediate node; and

code means for providing a time sensitive component of the file to the client from the originating node.

66. The computer program product of claim 65 further comprising:

code means for determining whether or not components of a file are time sensitive or time insensitive based on a comparison between a threshold value and a defined time value for each of the components.

67. The computer program product of claim 66 wherein the threshold value is dependent on the load of the originating server.

68. The computer program product of claim 65 further comprising the steps of:

code means for determining at the intermediate node whether or not the client is adapted to receive time insensitive and time sensitive components; and

if the client is not so adapted,

code means for assembling the file from the time sensitive and time insensitive components at the intermediate node; and

code means for transmitting the assembled file to the client from the intermediate node.

69. An apparatus for parsing a data file into its static and dynamic data components comprising:

means for setting a threshold value for separating static data components and dynamic data components;

means for obtaining expiration parameters for each data component of the data file;

means for comparing each expiration parameter to the threshold value;

means for identifying as dynamic data components the data components of the data file that have an expiration value lower than the threshold value; and

means for identifying as static data components the data components of the data file that have an expiration value higher than the threshold value.

70. The apparatus of claim 69 wherein the threshold value is set based on a measurement of a load value of a file server storing the data file.

71. The apparatus of claim 69 wherein the threshold value of a first file server is different than the threshold value of other file servers.

72. An apparatus for sending requested data to a client from an intelligent file server storing a file comprising:

means for receiving a file request from the client, the file including static data components and dynamic data components;

means for determining whether the client is intelligent;

means for dividing static data components into static data packets and dividing dynamic data components into dynamic data packets; and

means for transmitting the static data packets and the dynamic data packets to an intelligent client.

73. The apparatus of claim 72 wherein the intelligent file server stores in at least one memory location a copy of the requested file and a copy of the requested file divided into its static and dynamic components.

74. The apparatus of claim 72 wherein the intelligent file server stores in at least one memory location a copy of the requested file and creates a copy of the requested file divided into its static and dynamic components and create upon receiving the file request.

75. The apparatus of claim 72 further comprising:

means for identifying routes between the file server and the intelligent client that include at least one intelligent hardware; and

means for transmitting the static data components and the dynamic data components using the route that includes at least one intelligent hardware.

76. The apparatus of claim 72 further comprising:

means for counting the number of requests from intelligent clients for a particular data file;

means for broadcasting the static data components of said data file to each intelligent client once a number of requests for a particular file surpasses a set value.

77. The method of claim 76 wherein the static data components broadcasting further comprising:

(a) means for receiving a data file including static data components and dynamic data components;

(b) means for specifying a time interval;

(c) means for parsing the static data components into a plurality of static data blocks;

(d) means for determining a required number of time slots to send the data file, wherein each of said time slot has a duration substantially equal to said time interval;

(e) means for allocating to each time slot at least;

(1) a first of said plurality of data blocks; and

(2) optionally one or more additional data blocks, such that said plurality of data blocks is available to a client accessing said data file during any time slot; and

(f) means for sending said plurality of data blocks based on said allocating step (e).

78. An apparatus for processing data received from an intelligent server comprising:

means for sending a file request from an intelligent client to the intelligent server, the intelligent server storing a copy of a file divided into its static components and its dynamic components;

means for receiving static data packets formed by parsing the static data components and dynamic data packets formed by parsing the dynamic data components;

means for reconstructing the requested file from the received static data packets and dynamic data packets.

79. The apparatus of claim 78 including the step of receiving static data from a broadcast, further comprising:

(a) means for receiving a selection of a data file during a first time slot;

(b) means for receiving at least one data block of said data file during a second time slot; and

(c)during a next time slot:

(1) means for receiving any data block not already received;

(2) means for recombining the received data blocks to form the data file; and

(3) means for repeating step (c) until all data blocks of said data file have been received and recombined.

80. The apparatus of claim 79 wherein further comprising:

means for performing an error check; and means for sending a request to resend the data blocks generating the error check.

81. The apparatus of claim 79 wherein the step of recombining of the data blocks does not start until after all the data files have been received.

82. An apparatus for creating an intelligent protocol data packet comprising:

means for including identifying information identifying the intelligent data protocol.

83. The apparatus of claim 82 wherein said identification information is included in a header portion of the data packet.

84. The apparatus of claim 83 wherein the data packet is created based on TCP/IP protocol.

85. The apparatus of claim 84 wherein the header portion is the TCP information portion.

86. The apparatus of claim 82 wherein said identification information is included in the data portion of the data packet.

87. The apparatus of claim 86 wherein the data packet is created based on TCP/IP protocol.

88. An apparatus for servicing a data request to a file server from a client connected to the network through a network connection providing server comprising:

means for receiving a request for a file from the client directed to the file server;

means for forwarding the request to the file server;

means for receiving the data from the file server;

means for storing a copy of a static portions of the received data;

means for transmitting at least a dynamic data portion of the received data to the client.

89. The apparatus of claim 88 further comprising:

means for checking a memory related to the network connection providing server for the presence of a first group of data corresponding to the static portions of the requested file;

means for determining whether the file server is an intelligent file server;

means for sending the first group of data to the client;

means for sending a request to the file server for a second group of data not saved in the memory;

means for receiving the second group of data from the file server; and

means for sending at least the second group of data to the client.

90. An apparatus for responding to a file request from an intelligent client by an intelligent network connection providing server comprising:

means for receiving a file request from a client;

means for determining if the client is intelligent;

means for sending a static data to the intelligent client;

means for forwarding a request for at least a dynamic data related to the means for requested file to an intelligent server storing the requested file;

means for receiving at least the dynamic data ; and

means for transmitting at least the dynamic data to the client.

91. The apparatus for claim 90 wherein the client is not intelligent, further comprising:

means for recombining the received static data for the requested file and the received dynamic data for the requested file; and

means for repacketing the requested file before transmitting the requested file to the client.

92. The apparatus for claim 90 further comprising:

means for counting a number of requests from intelligent clients for a particular data file;

means for broadcasting the static data for the requested file once the number of requests for a particular file surpasses a set value.

93. The apparatus of claim 92 wherein broadcasting the static data further comprising:

(a) means for specifying a time interval;

(b) means for parsing the static data into a plurality of static data blocks;

(c) means for determining a required number of time slots to send the static data, wherein each of said time slot has a duration substantially equal to said time interval;

(d) means for allocating to each time slot at least;

(1) a first of the plurality of static data blocks; and

(2) optionally one or more additional static data blocks; and

(e) means for sending the plurality of static data blocks based on said allocating step (d).

94. An apparatus for the operation of a system for transmitting and receiving data within a network comprising:

means for dividing a file into dynamic data and static data by comparing at least one expiration parameter corresponding to each data part to a set threshold value;

means for transmitting a request for a file from an intelligent client;

means for receiving the file request by an intelligent server;

means for transmitting at least the dynamic data components of the requested file to the intelligent client; and

means for recombining the static components and the dynamic components of the requested file.

95. The apparatus of claim 94 further comprising:

means for identifying routes between the intelligent file server and the intelligent client that include at least one intelligent hardware; and

means for transmitting the static data components and the dynamic data components using the route that includes at least one intelligent hardware.

96. The apparatus of claim 94 further comprising:

means for counting the number of requests from intelligent clients for a particular data file;

means for broadcasting the static data components of said data file to each intelligent client once a number of request for a particular file surpasses a set value.

97. The apparatus of claim 96 wherein the static data broadcasting steps further comprises:

(a) means for receiving a static data blocks of a data file;

(b) means for specifying a time interval;

(c) means for parsing said data file into a plurality of data blocks based on said time interval such that each data block is displayable during said time interval;

(d) means for determining a required number of time slots to send said data file, wherein each of said time slot has a duration substantially equal to said time interval;

(e) means for allocating to each time slot at least;

(1) a first of said plurality of data blocks; and

(2) optionally one or more additional data blocks, such that said plurality of data blocks is available in sequential order to a client accessing said data file during any time slot; and

(f) means for sending said plurality of data blocks based on said allocating step (e).

98. The apparatus for claim 94 wherein the intelligent file server stores a copy of the requested file separated into its static data components and its dynamic data components and a copy of the requested file that is not separated into its dynamic data components and static data components.

99. An apparatus for sending a file over a network from an originating node to a requesting client, comprising:

means for receiving a request for the file at an intermediate node;

means for providing a time insensitive component of the file to the client from the intermediate node; and

means for providing a time sensitive component of the file to the client from the originating node.

100. The apparatus of claim 99 further comprising:

means for determining whether or not components of a file are time sensitive or time insensitive based on a comparison between a threshold value and a defined time value for each of the components.

101. The apparatus of claim 100 wherein the threshold value is dependent on the load of the originating server.

102. The apparatus of claim 99 further comprising the steps of:

means for determining at the intermediate node whether or not the client is adapted to receive time insensitive and time sensitive components; and

if the client is not so adapted,

means for assembling the file from the time sensitive and time insensitive components at the intermediate node; and

means for transmitting the assembled file to the client from the intermediate node.

103. A broadcast server method of transmitting data files, comprising the acts of:

providing a data file consisting of static data and dynamic data;

parsing said data file into a static data component and at least one dynamic data component;

transmitting said static data component and said at least one dynamic component via a transmission medium such that said static component and said at least one dynamic component may be displayed visually; and

transmitting at least one additional dynamic component that supercedes said at least one dynamic component via said transmission medium such that said at least one additional dynamic component may be displayed with said static component, wherein said additional dynamic component is displayed in place of said at least one dynamic component.

104. A method as recited in claim 103, wherein said act of parsing said data file includes determining whether each component of said data file is a static component or a dynamic component.

105 A method as recited in claim 104, wherein each component of said data file has an expiration value, and wherein said determination is made in response to said expiration value.

106. A method as recited in claim 105, wherein components having an expiration value greater than a predetermined threshold value are determined to be static

components and components having an expiration value less than said predetermined threshold value are determined to be dynamic components.

107. A method as recited in claim 105, wherein said static components are divided into static data packets and dynamic components are divided into dynamic data packets for transmission.

108. A method as recited in claim 107, wherein each data packet has a header containing information indicating whether said data packet is a static data packet or a dynamic data packet.

109. A method as recited in claim 105, wherein said static and dynamic data components are transmitted in a scheduling sequence at least partially responsive to said expiration values.

110. A method as recited in claim 109, wherein data components having higher expiration values are transmitted more often than data components having lower expiration values.

111. A method as recited in claim 109, wherein said static data is transmitted to a plurality of users, and wherein said additional dynamic data component is transmitted to a subgroup of said plurality of users, thereby providing a display having a generic static component and a dynamic component unique to said subgroup.

112. A method for displaying data files having static and dynamic components received from a broadcast server, comprising the acts of:

receiving static data blocks containing portions of a static image;

receiving dynamic data blocks containing portions of a plurality of dynamic images;

combining said static and dynamic images over a plurality of time intervals, wherein at each time interval said static image is combined with a dynamic image, thereby forming a display having both dynamic and static components.

113. A method as recited in claim 112, wherein said received static image is stored in a first memory location and displayed from said memory location.

114. A method as recited in claim 112, further comprising the acts of:

sending a file request to a broadcast server; and

receiving said file parsed as said dynamic and static data blocks.

115. A method as recited in claim 112, wherein said dynamic images are stored in a second memory location and displayed from memory, wherein the most recently received dynamic images are displayed.

116. A method as recited in claim 115, wherein said static images are stored in a first memory location and displayed from memory, wherein the most recently received static image is displayed.